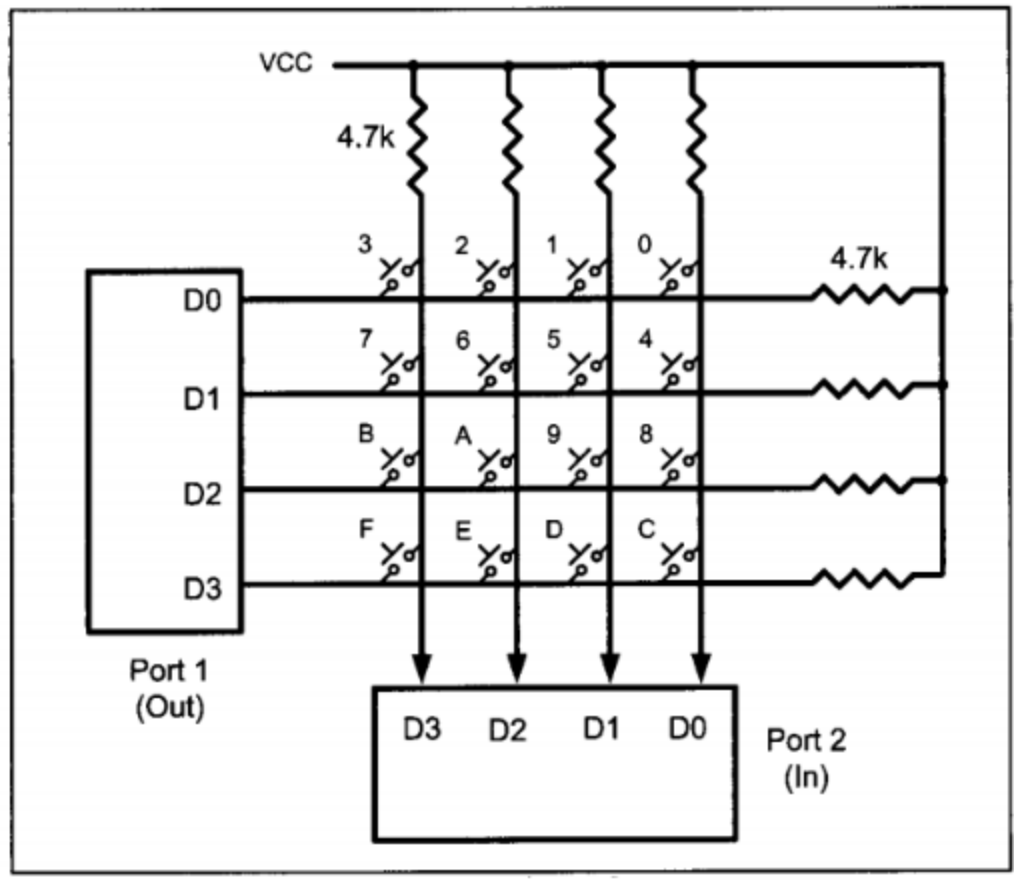
Microcontroller Applications

Architecture:

- Basic ARM architecture, ARM organization core Data flow Model,
- ARM register organization, current program register organization.
- Pin configuration and architecture Arduino,
- Introduction to Raspberry Pi, Understanding SoC architecture and SoCs used in
- Raspberry Pi, Pin Description of Raspberry Pi,
- On-board components of Rpi
- Microcontroller Applications: Interfacing matrix keyboard and
- Seven segments LED display,
- LCD Interfacing,
- ADC Interfacing,
- DC motor interfacing.

Interfacing the Keyboard with AVR Microcontroller

- At a lowest level, keyboards are organized in a matrix of rows and columns. The CPU accesses both rows and columns through ports, therefore with two 8-bit ports an 8 X 8 matrix of keys can be connected with a microcontroller.
- When a key is pressed, a column and a row make a contact; otherwise, there is no connection between rows and columns. In x86 PC keyboards, a single microcontroller use for software and hardware interfacing of the keyboard.
- The matrix keyboard connection with the port 1 and port 2 of AVR microcontroller is shown below:


VCC
4.7k
3
2
1
0
4.7k
D0
7
6
5
4
D1
B
A
9
8
D2
F
E
D
C
D3
Port 1
(Out)
D3
D2
D1
D0
Port 2
(In)

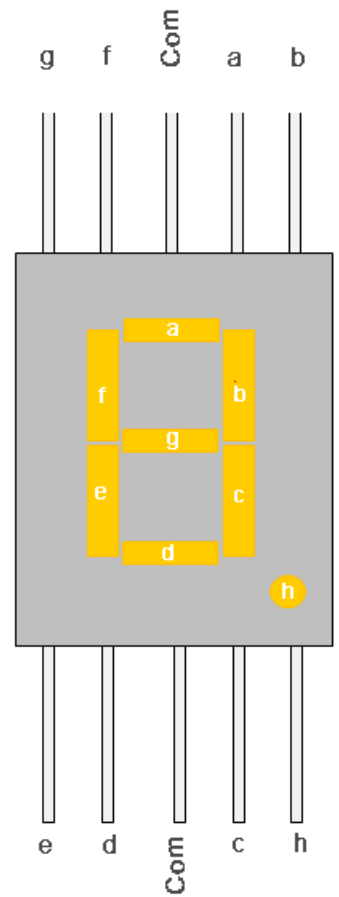
- Scanning and identifying the key:
- Above figure represent a 4 X 4 matrix connected to two ports.
- The rows are connected to an output port and the columns are connected to an input port.
- If no key has been pressed, reading the input port will give 1s for all columns since they are connected to high (VCC).
- If all the rows are grounded and key is pressed, one of the columns have 0 since the key is pressed provides the path to ground.
- It is the function of microcontroller to scan the keyboard continuously to detect and identify the key pressed.

- ## Grounding rows and reading columns:

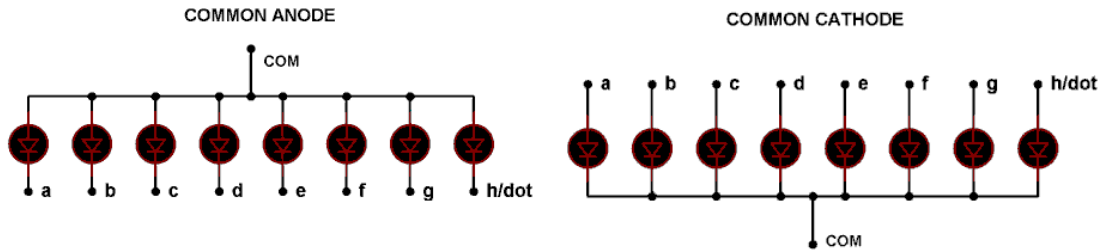
- For detecting the pressed key, the microcontroller grounds all rows by providing 0 to the input latch, and then it reads columns. If the data read from the columns is D3-D0=1111, no key has been pressed and the process continues until a key pressed is detected by system.
- However, if one of the column bits has a zero, this means that a key press has occurred. For example, if D3-D0= 1101, it means that a key in the D1 column has been pressed. After a key press is detected, the microcontroller will pass through the process of identifying the key.
- Starting with the top row, the microcontroller grounds it for providing a low to row D0 only; then it reads the columns.
- If the data read is all 1s, no key in a row is activated and the process is further moved to the next row.
- The process of identifying the rows and columns is easy as microcontroller knows at any time which rows and columns are being accessed.

- 2. Seven segments LED display,

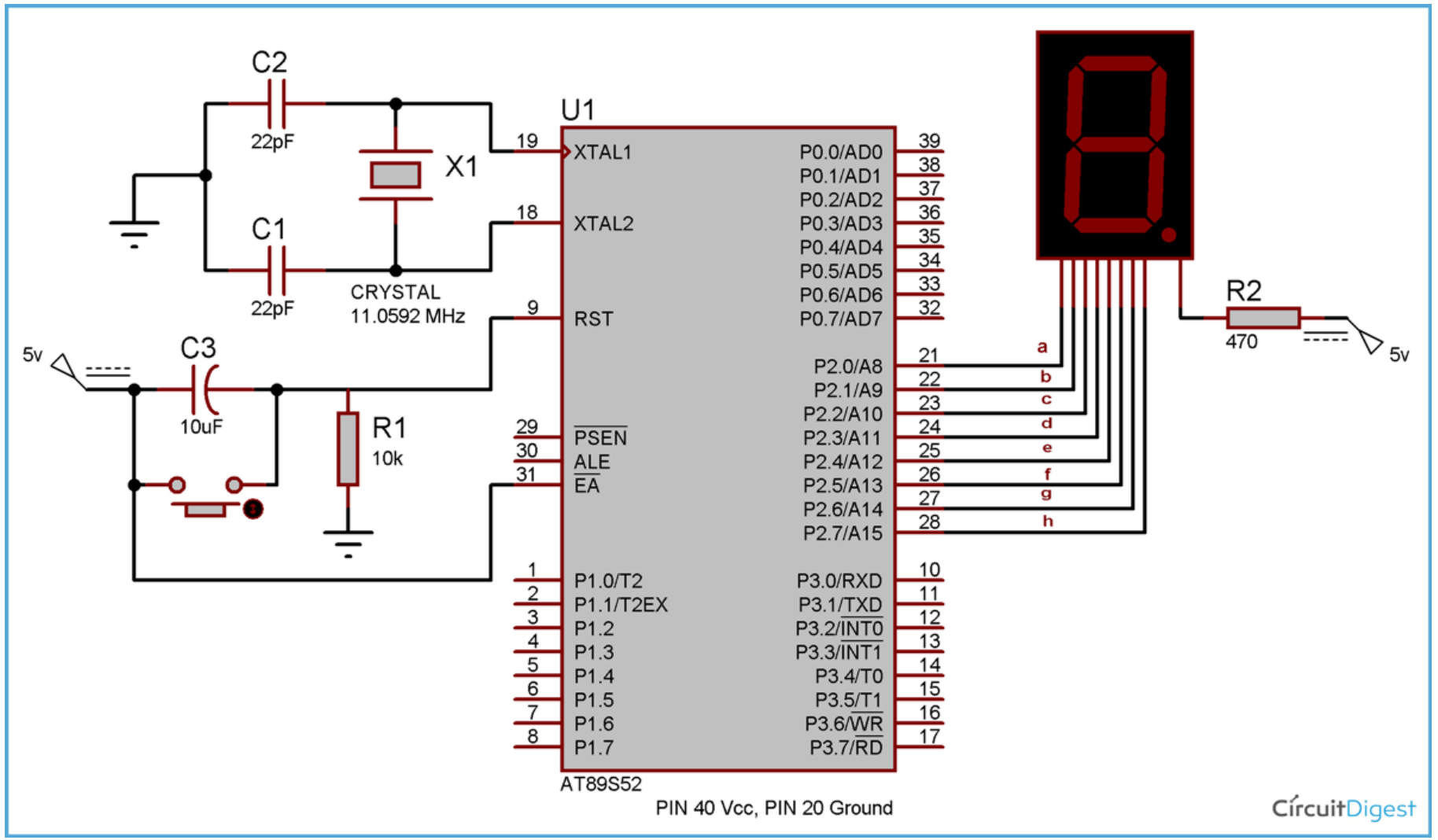
- Seven segment displays are important display units in Electronics and widely used to display numbers from 0 to 9.
- It can also display some character alphabets like A,B,C,H,F,E etc.
- It's the simplest unit to display numbers and characters.
- It just consists 8 LEDs, each LED used to illuminate one segment of unit and the 8th LED used to illuminate DOT in 7 segment display.
- We can refer each segment as a LINE, as we can see there are 7 lines in the unit, which are used to display a number/character.
- We can refer each line/segment "a,b,c,d,e,f,g" and for dot character we will use "h".
- There are 10 pins, in which 8 pins are used to refer a,b,c,d,e,f,g and h/dp, the two middle pins are common anode/cathode of all he LEDs. These common anode/cathode are internally shorted so we need to connect only one COM pin.


g
f
Com
a
b
a
f
b
g
e
c
d
h
e
d
Com
c
h

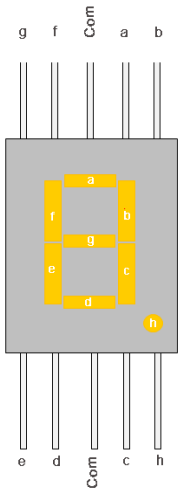
- There are two types of 7 segment displays: Common Anode and Common Cathode:
- **Common Anode:** In this all the Negative terminals (cathode) of all the 8 LEDs are connected together (see diagram below), named as COM. And all the positive terminals are left alone.
- **Common Cathode:** In this all the positive terminals (Anodes) of all the 8 LEDs are connected together, named as COM. And all the negative thermals are left alone.



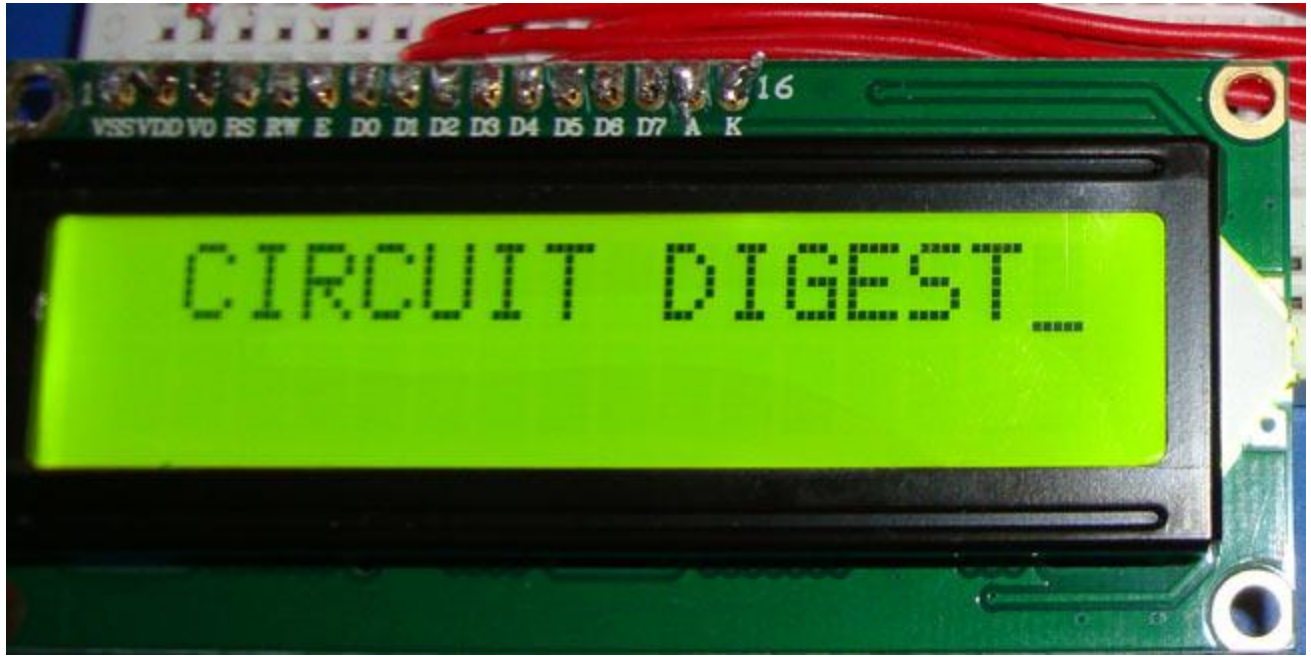
Internal connections of 7 Segment Display


C2
22pF
X1
C1
22pF
CRYSTAL
11.0592 MHz
U1
19
XTAL1
18
XTAL2
9
RST
29
PSEN
30
ALE
31
EA
1
P1.0/T2
2
P1.1/T2EX
3
P1.2
4
P1.3
5
P1.4
6
P1.5
7
P1.6
8
P1.7
AT89S52
P0.0/AD0
P0.1/AD1
P0.2/AD2
P0.3/AD3
P0.4/AD4
P0.5/AD5
P0.6/AD6
P0.7/AD7
P2.0/A8
P2.1/A9
P2.2/A10
P2.3/A11
P2.4/A12
P2.5/A13
P2.6/A14
P2.7/A15
P3.0/RXD
P3.1/TXD
P3.2/INT0
P3.3/INT1
P3.4/T0
P3.5/T1
P3.6/WR
P3.7/RD
39
38
37
36
35
34
33
32
21
22
23
24
25
26
27
28
10
11
12
13
14
15
16
17
a
b
c
d
e
f
g
h
R2
470
5v
5v
C3
10uF
R1
10k
PIN 40 Vcc, PIN 20 Ground
CircuitDigest

- As shown above the circuit diagram for **interfacing 7 segment display with 8051 microcontroller**, we have connected a,b,c,d,e,f,g,h to pins 2.0 to 2.7 means we are connecting 7 segment to port 2 of microcontroller.
- Now suppose we want to display 0, then we need to glow all the LEDs except LED which belongs to line "g" , so pins 2.0 to 2.6 should be at 0 (should be 0 to TURN ON the LED as per negative logic) and pin 2.7 and 2.8 should be at 1 (should be 1 to TURN OFF the LED as per negative logic). So the LEDs connected to pins 2.0 to 2.6 (a,b,c,d,e,f) will be ON and LEDs connected to 2.7 and 2.8 (g and h) will be OFF, that will create a "0" in 7 segment.
- So we need bit pattern 11000000 (Pin 8 is the highest bit so starting from P2.7 to P2.0 EX code for binary 11000000 is "C0".
- Similarly we can calculate for all the digits. Here we should note that we are keeping "c s OFF, so we need to give LOGIC "1" to it every time. A table has been given below for all the nu using Common Anode 7 segment.



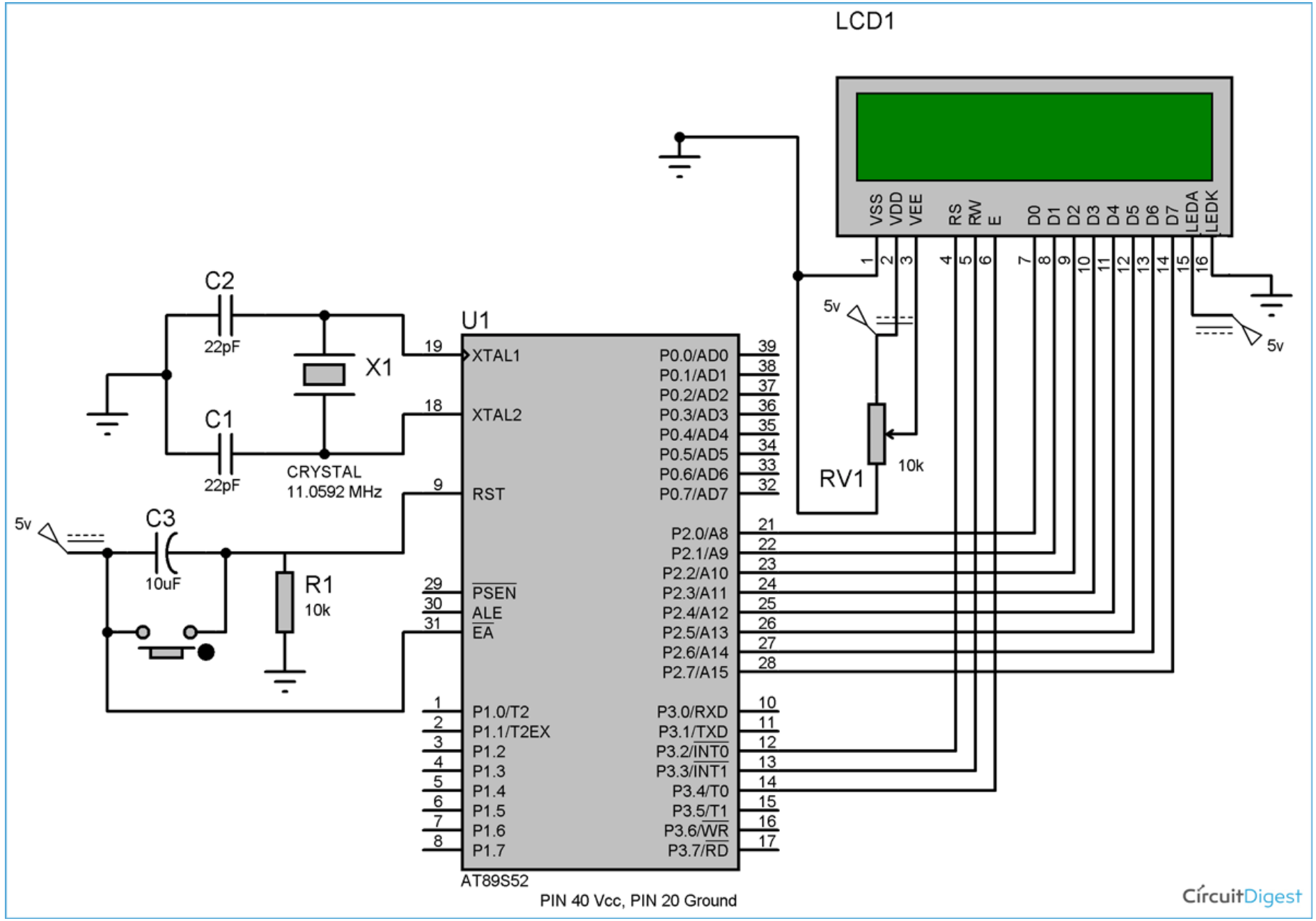
3. LCD Interfacing with 8051 Microcontroller



- Display units are the most important output devices in embedded projects and electronics products.
- 16x2 LCD is one of the most used display unit.
- 16x2 LCD means that there are two rows in which 16 characters can be displayed per line, and each character takes 5X7 matrix space on LCD.

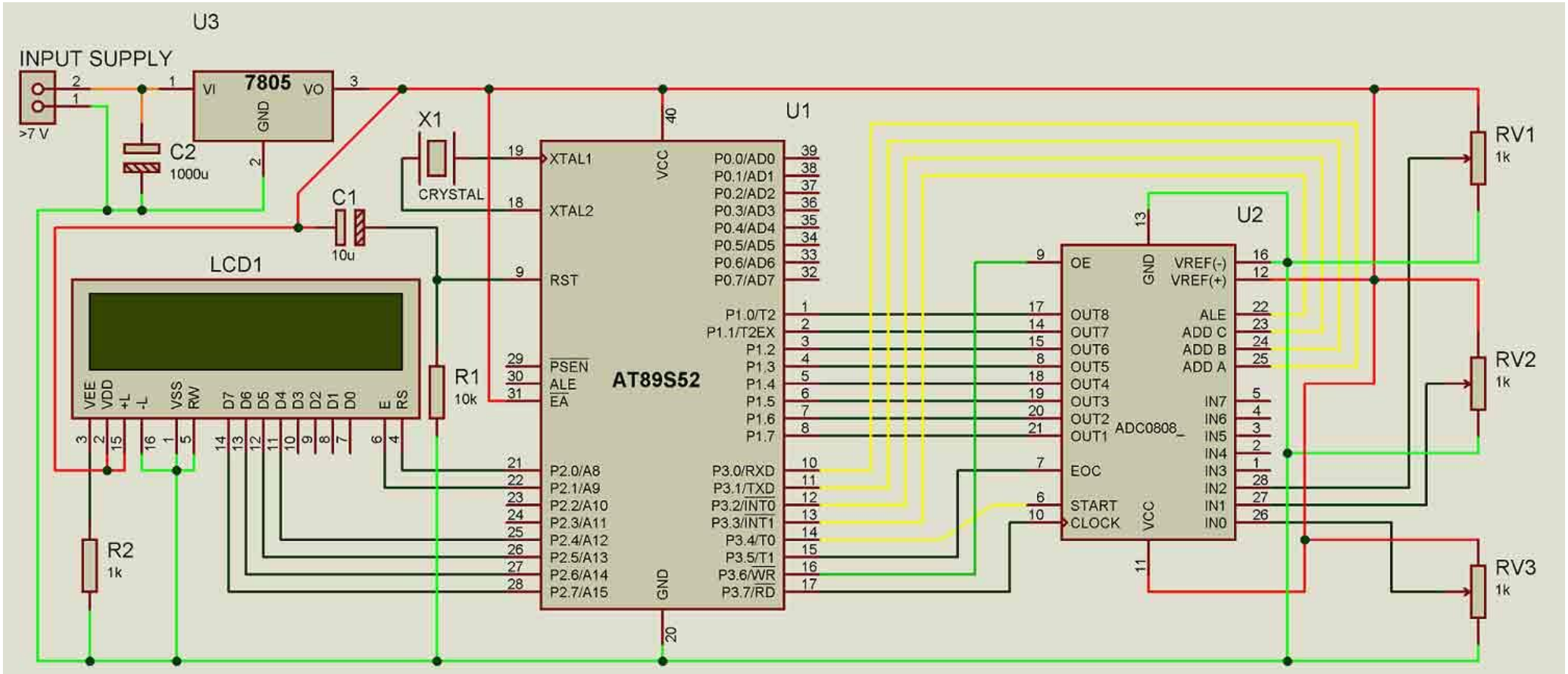
Category	Pin NO.	Pin Name	Function
Power Pins	1	VSS	Ground Pin, connected to Ground
	2	VDD or Vcc	Voltage Pin +5V
Contrast Pin	3	V0 or VEE	Contrast Setting, connected to Vcc thorough a variable resistor.
Control Pins	4	RS	Register Select Pin, RS=0 Command mode, RS=1 Data mode
	5	RW	Read/ Write pin, RW=0 Write mode, RW=1 Read mode
	6	E	Enable, a high to low pulse need to enable the LCD
Data Pins	7-14	D0-D7	Data Pins, Stores the Data to be displayed on LCD or the command instructions
Backlight Pins	15	LED+ or A	To power the Backlight +5V
	16	LED- or K	Backlight Ground

- All the pins are clearly understandable by their name and functions, except the control pins, so they are explained below:
- **RS:** RS is the register select pin. We need to set it to 1, if we are sending some data to be displayed on LCD. And we will set it to 0 if we are sending some command instruction like clear the screen (hex code 01).
- **RW:** This is Read/write pin, we will set it to 0, if we are going to write some data on LCD. And set it to 1, if we are reading from LCD module. Generally this is set to 0, because we do not have need to read data from LCD. Only one instruction "Get LCD status", need to be read some times.
- **E:** This pin is used to enable the module when a high to low pulse is given to it. A pulse of 450 ns should be given. That transition from HIGH to LOW makes the module ENABLE.


LCD1
VSS
VDD
VEE
RS
RW
E
D0
D1
D2
D3
D4
D5
D6
D7
LEDA
LEDK
5v
RV1
10k
C2
22pF
C1
22pF
X1
CRYSTAL
11.0592 MHz
U1
XTAL1
XTAL2
RST
PSEN
ALE
EA
P0.0/AD0
P0.1/AD1
P0.2/AD2
P0.3/AD3
P0.4/AD4
P0.5/AD5
P0.6/AD6
P0.7/AD7
P2.0/A8
P2.1/A9
P2.2/A10
P2.3/A11
P2.4/A12
P2.5/A13
P2.6/A14
P2.7/A15
P1.0/T2
P1.1/T2EX
P1.2
P1.3
P1.4
P1.5
P1.6
P1.7
P3.0/RXD
P3.1/TXD
P3.2/INT0
P3.3/INT1
P3.4/T0
P3.5/T1
P3.6/WR
P3.7/RD
C3
10uF
R1
10k
AT89S52
PIN 40 Vcc, PIN 20 Ground
CircuitDigest

4. Interfacing ADC0808 with 8051 Microcontroller

- **ADC is the Analog to Digital converter**, which converts analog data into digital format; usually it is used to convert **analog voltage** into digital format.
- Analog signal has infinite no of values like a sine wave or our speech, ADC converts them into particular levels or states, which can be measured in numbers as a physical quantity.
- Instead of continuous conversion, ADC converts data periodically, which is usually known as sampling rate. **Telephone modem** is one of the examples of ADC, which is used for internet, it converts analog data into digital data, so that computer can understand, because computer can only understand Digital data.
- The major advantage, of using ADC is that, we noise can be efficiently eliminated from the original signal and digital signal can travel more efficiently than analog one. That's the reason that digital audio is very clear, while listening.
- In present time there are lots of microcontrollers in market which has inbuilt ADC with one or more channels. And by using their ADC register we can interface
- . When we select **8051 microcontroller** family for making any project, in which we need of an ADC conversion, then we use **external ADC**. Some external ADC chips are 0803,0804,0808,0809 and there are many more.

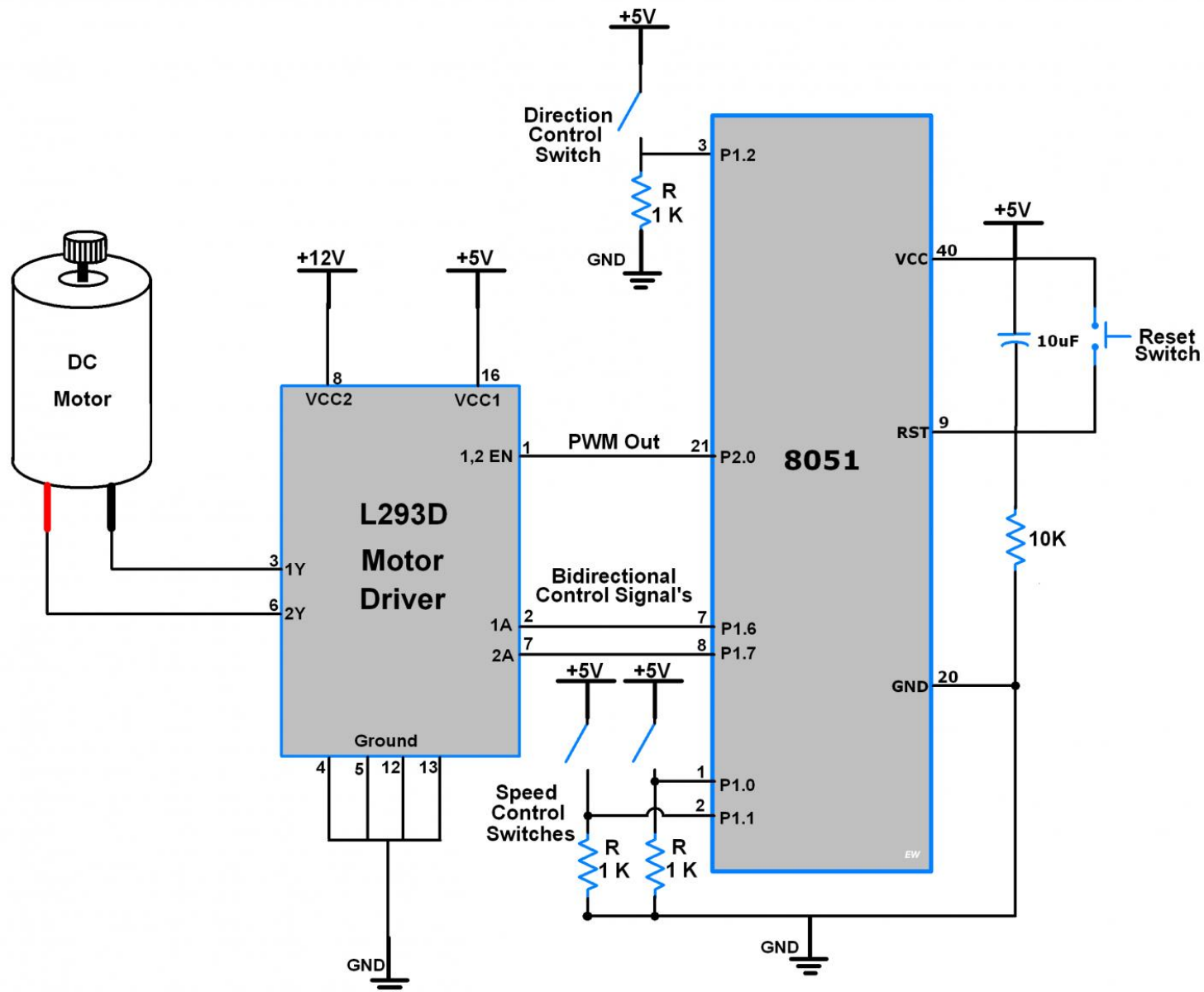

U3
INPUT SUPPLY
>7 V
7805
VI
VO
GND
C2
1000u
C1
10u
X1
CRYSTAL
U1
AT89S52
XTAL1
XTAL2
RST
PSEN
ALE
EA
VCC
GND
P0.0/AD0
P0.1/AD1
P0.2/AD2
P0.3/AD3
P0.4/AD4
P0.5/AD5
P0.6/AD6
P0.7/AD7
P1.0/T2
P1.1/T2EX
P1.2
P1.3
P1.4
P1.5
P1.6
P1.7
P2.0/A8
P2.1/A9
P2.2/A10
P2.3/A11
P2.4/A12
P2.5/A13
P2.6/A14
P2.7/A15
P3.0/RXD
P3.1/TXD
P3.2/INT0
P3.3/INT1
P3.4/T0
P3.5/T1
P3.6/WR
P3.7/RD
LCD1
VEE
VDD
+L
-L
VSS
RW
D7
D6
D5
D4
D3
D2
D1
D0
E
RS
R1
10k
R2
1k
U2
ADC0808_
OE
GND
VREF(-)
VREF(+)
OUT8
OUT7
OUT6
OUT5
OUT4
OUT3
OUT2
OUT1
EOC
START
CLOCK
VCC
ALE
ADD C
ADD B
ADD A
IN7
IN6
IN5
IN4
IN3
IN2
IN1
IN0
RV1
1k
RV2
1k
RV3
1k

- ## Working:

- When we power the circuit then microcontroller initialize the LCD by using appropriate command, gives clock to ADC chip,
- selects ADC channel by using address line and send start conversion signal to ADC.
- After this ADC first reads selected ADC channel input and gives its converted output to microcontroller.
- Then microcontroller shows its value at Ch1 position in LCD. And then microcontroller changes ADC channel by using address line.
- And then ADC reads selected channel and send output to microcontroller. And show on LCD as name Ch2. And like wise for other channels.

5. Interfacing DC Motor with 8051

- DC motor converts electrical energy in the form of Direct Current into mechanical energy.
- In the case of the motor, the mechanical energy produced is in the form of a rotational movement of the motor shaft.
- The direction of rotation of the shaft of the motor can be reversed by reversing the direction of Direct Current through the motor.
- The motor can be rotated at a certain speed by applying a fixed voltage to it. If the voltage varies, the speed of the motor varies.
- Thus, the DC motor speed can be controlled by applying varying DC voltage; whereas the direction of rotation of the motor can be changed by reversing the direction of current through it.
- For applying varying voltage, we can make use of the PWM technique.
- For reversing the current, we can make use of an H-Bridge circuit or motor driver ICs that employ the H-Bridge technique or any other mechanisms.


+5V
Direction
Control
Switch
3
P1.2
R
1 K
GND
+5V
VCC
40
+12V
+5V
10uF
Reset
Switch
DC
Motor
8
16
VCC2
VCC1
RST
9
1,2 EN
1
PWM Out
21
P2.0
8051
L293D
Motor
Driver
10K
3
1Y
Bidirectional
Control Signal's
6
2Y
1A
2
7
P1.6
2A
7
8
P1.7
+5V
+5V
GND
20
Ground
4
5
12
13
1
P1.0
Speed
Control
Switches
2
P1.1
R
1 K
R
1 K
GND
GND

- As shown in the above figure we have connected two toggle switches on the P1.0 and P1.1 pin of the AT89S52 microcontroller to change the speed of the DC motor by 10%.
- One toggle switch at pin P1.2 controls the motor's rotating direction.
- P1.6 and P1.7 pins are used as output direction control pins. It provides control to motor1 input pins of the L293D motor driver which rotate the motor clockwise and anticlockwise by changing their terminal polarity.
- And Speed of the DC Motor is varied through PWM Out pin P2.0.
- Here we are using the timer of AT89S52 to generate PWM.

Thank you